UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/716,840	11/18/2003	Matthew Marcus	07844-619001	7089

21876 7590 03/08/2007

FISH & RICHARDSON P.C.
P.O. Box 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
SAEED, USMAAN	
ART UNIT	**PAPER NUMBER**
2166	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/08/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	**Application No.** 10/716,840	**Applicant(s)** MARCUS, MATTHEW
	Examiner Usmaan Saeed	**Art Unit** 2166

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 December 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-34 and 37-40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-34 and 37-40* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *18 November 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. Receipt of Applicant's Amendment, filed on 12/07/2006 is acknowledged. Claims 1, 3, 10-11, 20, and 29-30 have been amended. Claims 35-36 have been cancelled.

Claim Objections

2. The amendments to claims 35 and 36 were received on 12/07/2006 and are acceptable to overcome the objections.

Claim Rejections - 35 USC § 101

3. The amendments to the claims were received on 12/07/2006 and are acceptable to overcome some 101 rejections.

Claims 20-40 are still rejection because the computer readable medium recited in these claims also includes information carries, which are not tangible mediums. Applicant is suggested to write "computer readable storage medium" instead of "computer readable medium".

Specification

4. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: The word “tangible” in claim 20 is not present in the specification.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1- 34 and 37-40 are rejected under 35 U.S.C. 102(b) as being anticipated by **Chau et al.** (**Chau** hereinafter) (U.S. PGPub No. 2002/0156772).

With respect to claim 1, **Chau** teaches **a method for searching for one or more logical elements in a hierarchical tree structure of an extensible markup language XML document conforming to a schema used for XML, comprising:**

"providing a representation of an extensible markup language XML document instance containing two or more logical elements, wherein at least one logical element is a parent node and at least one logical element is a child node in a hierarchical tree structure describing the representation" as XML Path or XPath addresses parts of an XML document. XPath gets its name from its use of a path notation as in URLs for navigating through the hierarchical structure of an XML document (**Chau** Paragraph 0042). Represents the element with the name tagn, which is a child of the descending chain from root, tag1, tag2, . . . where tag3 has the value "Los Angeles" (**Chau** Paragraph 0112).

"receiving a query for logical elements satisfying an Xpath expression" as the XML System also allows overrides of query conditions explicitly or implicitly defined in the DAD, by parsing the SQL or XML XPath based override parameter to the composition stored procedures (**Chau** Paragraph 0080).

"searching in the hierarchical tree structure only nodes that potentially have child nodes satisfying the Xpath expression" as the side tables are created by the DAD, and indices are created for columns in the side tables. Therefore, the search will be fast with indexing. Note that the invoice_number is the primary key in the application table sales_tab. The advantage of direct query with sub-query is better performance. When side tables have parent-children relationships, direct query with sub-query often make more sense (**Chau** Paragraph 0335 & 0336). Represents the element with the name tagn, which is a child of the descending chain from root, tag1, tag2, . . . where tag3 has the value "Los Angeles" (**Chau** Paragraph 0112). The FROM clause defines the tables containing the data, and the WHERE clause specifies the join and search conditions (**Chau** Paragraph 0679). Examiner interprets that every node in this reference has a potential child nodes.

"providing the logical elements satisfying the XPath expression" as XML System provides extracting UDFs to retrieve XML elements or attributes in the SQL select clause. This is very useful after search filtering on a collection of XML documents to further obtain desired elements or attributes (**Chau** Paragraph 0318).

Claim 20 is essentially the same as claim 1 except that it sets forth the claimed invention as a computer program product and is rejected for the same reason as applied hereinabove.

With respect to claims 2 & 3, **Chau** teaches **"the method of claim 1, including the further step of generating a collection of parent nodes that potentially have child nodes satisfying the Xpath expression from a table relating a class of parent nodes/parent nodes and a class of child nodes/child nodes, and wherein the table is used in the final searching step"** as for the root element_node, all tables storing its attribute or all child element data should be specified (**Chau** Paragraph 0164). Representing an XML element. It must be defined in the specified DTD. For the RDB_node mapping, the root element_node must have a RDB_node to specify all tables containing XML data for itself and all its children nodes. It can have zero or more attribute_nodes and child element_nodes, as well as zero or one text_node (**Chau** Paragraph 0176).

Claims 21 & 22 are essentially the same as claims 2 & 3 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claims 4 & 5, **Chau** teaches **"the method of claim 2, wherein the table comprises entries containing hash representations of a class of parent nodes/parent nodes and a class of child nodes/child nodes"** as levelmap is an associative array that maps column names to their equivalence class numbers or "relational level." The equivalence classes in ascending order of relational levels should have one-to-many relationship between each adjacent classes with the "many" side at the upper level. In the example, "order_key" maps to 0; "part_key" maps to 1; and "date" maps to 2. The associative array can be implemented in memory, for example, as a hash table, a sorted array, or a binary search tree (**Chau** Paragraph 0699). Create a hash table or stored array of all top_elements rel->top_elements for fast search (**Chau** Paragraph 0941).

Claims 23 & 24 are essentially the same as claims 4 & 5 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 6 & 7, **Chau** teaches **"the method of claim 2, wherein the table comprises a listing of permitted classes of child nodes/child nodes for each class of parent node/parent node"** as for the root element_node, all tables storing its attribute or all child element data should be specified (**Chau** Paragraph 0164). Representing an XML element. It must be defined in the specified DTD. For the RDB_node mapping, the root element_node must have a RDB_node to specify all tables containing XML data for itself and all its children nodes. It can have zero or more attribute_nodes and child element_nodes, as well as zero or one text_node (**Chau** Paragraph 0176 & 0119). Tables contain lists of data, therefore this reference contain table with lists of root/parent nodes and children nodes. Examiner refers these non restricted paths as permitted paths.

Claims 25 & 26 are essentially the same as claims 6 & 7 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 8 & 9, **Chau** teaches **"the method of claim 2, wherein the table comprises a listing of permitted classes of parent nodes/parent nodes for each class of child node/child node"** as for the root element_node, all tables storing its attribute or all child element data should be specified (**Chau** Paragraph 0164). Representing an XML element. It must be defined in the specified DTD. For the

RDB_node mapping, the root element_node must have a RDB_node to specify all tables containing XML data for itself and all its children nodes. It can have zero or more attribute_nodes and child element_nodes, as well as zero or one text_node (**Chau** Paragraph 0176 & 0119). Tables contain lists of data, therefore this reference contain table with lists of root/parent nodes and children nodes. Examiner refers these non restricted paths as permitted paths.

Claims 27, and 28 are essentially the same as claims 8 & 9 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 10 & 11, **Chau** teaches **"the method of claim 1, further comprising:**

"receiving a rule set identifying allowable combinations between child nodes and parent nodes in a hierarchical document structure" as Extensible Markup Language (XML) is a set of rules or guidelines for designing text formats for structured data using tags (**Chau** Paragraph 0040). Represents the element with the name tagn which is a child of the descending chain from root, tag1, tag2, . . . where tag3 has the value "Los Angeles" (**Chau** Paragraph 0112 & 0119). Examiner refers these non-restricted paths as permitted paths and these permitted paths contain allowable combinations.

"transforming the rule set into a table relating a class of parent nodes/parent nodes and a class of child nodes/child nodes" as the element with the name tagn which is a child of the descending chain from root, tag1, tag2, . . . where tag3 has the value "Los Angeles" (**Chau** Paragraph 0112 & 0119). A transformation expressed in XSLT describes rules for transforming a source tree into a result tree (**Chau** Paragraph 0041).

"using the table in the searching step" as the side tables are created by the DAD, and indices are created for columns in the side tables. Therefore, the search will be fast with indexing. Note that the invoice_number is the primary key in the application table sales_tab. The advantage of direct query with sub-query is better performance. When side tables have parent-children relationships, direct query with sub-query often make more sense (**Chau** Paragraph 0335 & 0336).

Claims 29 & 30 are essentially the same as claims 10 & 11 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 12 & 13, **Chau** teaches **"the method of claim 10, wherein: the rule set includes one of: an XML schema, a DTD, and a RelaxNg schema"** as Extensible Markup Language (XML) is a set of rules or guidelines for designing text formats for structured data using tags (**Chau** Paragraph 0040).

Claims 31 & 32 are essentially the same as claims 12 & 13 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 14 & 15, **Chau** teaches **"the method of claim 2, wherein the table includes a listing of a not-permitted class of child nodes/child nodes for each class of parent node/parent node"** as there are restrictions on the location path when used by the XML systems, and these are listed in the table below (**Chau** Paragraph 0115). The location path includes both parents and child nodes. Therefore the restricted path will not permit the listing of it parent and child nodes.

Claims 33 & 34 are essentially the same as claims 14 & 15 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 16 & 17, **Chau** teaches **"the method of claim 2, wherein the table includes a listing of a not-permitted class of parent nodes/parent nodes for each class of child node/child node"** as there are restrictions on the location path when used by the XML systems, and these are listed in the table below (**Chau** Paragraph 0115). The location path includes both parents and child nodes. Therefore the restricted path will not permit the listing of it parent and child nodes.

Claims 37 & 38 are essentially the same as claims 16 & 17 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 18 & 19 **Chau** teaches **the method of claim 1, further comprising the additional steps of:**

"receiving a rule set identifying non-allowable combinations between child nodes and parent nodes in a hierarchical document structure" as Extensible Markup Language (XML) is a set of rules or guidelines for designing text formats for structured data using tags (**Chau** Paragraph 0040). There are restrictions on the location path when used by the XML systems, and these are listed in the table below (**Chau** Paragraph 0115). The location path includes combination of parents and child nodes. Therefore the restricted path will not permit the any combination of it parent and child nodes.

"transforming the rule set into a table relating a class of parent nodes/parent node and a class of child nodes/child nodes" as a transformation expressed in XSLT describes rules for transforming a source tree into a result tree (**Chau** Paragraph 0041). The term simple location path refers to the c and f notations in the table for Restriction of Location Path Supported. The simple location path is a sequence of element type names connected by the "/" notation. Each element type may be qualified by its attribute values (**Chau** Paragraph 0117).

Claims 39 & 40 are essentially the same as claims 18 & 19 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

Response to Arguments

6. Applicant's arguments filed 12/07/2006 have been fully considered but they are not persuasive. A detailed discussion is set forth herein below.

Regarding claim 1, applicant argues that **Chau** does not teaches, **"providing a representation of an XML document instance"** and says that in Chau's Xpath are not used to represent XML documents.

In response to the preceding argument, Examiner respectfully submits that **Chau** teaches **"providing a representation of an XML document instance"** as XML Path or XPath addresses parts of an XML document. XPath gets its name from its use of a path notation as in URLs for navigating through the hierarchical structure of an XML document (**Chau** Paragraph 0042). Therefore the reference has representations of XML documents.

Further, **Chau** teaches a technique is provided for generating one or more XML documents from a relational database using the XPath data model (**Chau** Abstract). These lines teach that Xpath model is being used in generating/representing XML documents.

It is noted that Xpath are not required by the claimed limitation for representation of XML documents.

Further applicant argues that **Chau** does not teach **"a hierarchical tree structure describing the representation"** and say that a descending chain of **Chau** is not a hierarchical tree structure.

In response to the preceding argument, Examiner respectfully submits that **Chau** teaches **"a hierarchical tree structure describing the representation"** as XML Path or XPath addresses parts of an XML document. XPath gets its name from its use of a path notation as in URLs for navigating through the hierarchical structure of an XML document (**Chau** Paragraph 0042). Represents the element with the name tagn which is a child of the descending chain from root, tag1, tag2, . . . where tag3 has the value "Los Angeles" (**Chau** Paragraph 0112).

Further, **Chau** teaches initially, a document object model tree is generated using a document access definition, which defines the mapping between an XML tree structure and relational tables. The document object model tree is traversed to obtain information to retrieve relational data (**Chau** Abstract).

Further applicant argues that **Chau** does not teach, **"searching in the hierarchical tree structure only nodes that potentially have child nodes satisfying the Xpath expression."**

In response to the preceding argument, Examiner respectfully submits that **Chau** teaches **"searching in the hierarchical tree structure only nodes that potentially have child nodes satisfying the Xpath expression"** as the side tables are created by the DAD, and indices are created for columns in the side tables. Therefore, the search will be fast with indexing. Note that the invoice_number is the primary key in the application table sales_tab. The advantage of direct query with sub-query is better performance. When side tables have parent-children relationships, direct query with sub-query often make more sense (**Chau** Paragraph 0335 & 0336). Represents the element with the name tagn, which is a child of the descending chain from root, tag1, tag2, . . . where tag3 has the value "Los Angeles" (**Chau** Paragraph 0112). The FROM clause defines the tables containing the data, and the WHERE clause specifies the join and search conditions (**Chau** Paragraph 0679). Examiner interprets that every node in this reference has a potential child nodes.

Further, **Chau** discloses initially, a document object model tree is generated using a document access definition. The document object model tree is traversed to obtain information to retrieve relational data. The relational data is mapped to one or more XML documents (**Chau** Paragraph 0018). Initially, an XML document containing XML data is received. A document access definition that identifies one or more relational tables and columns is received. The XML data is mapped from the application DTD to the relational tables and columns using the document access definition based on the XPath data model (**Chau** Paragraph 0019). The XML System gives applications the freedom to specify a list of XML elements/attributes as general

SQL data types for fast search. The XML System will extract these values from the XML documents and store them in side tables (**Chau** Paragraph 0075).

Therefore the reference teaches that the side tables are made up of nodes in a hierarchical tree structure because the relational side tables contain elements or attributes of XML documents and there is mapping between an XML tree structure and relational tables.

Further figure 10 shown below also teaches tree comprises relational database nodes.

Patent Application Publication Oct. 24, 2002 Sheet 10 of 11 US 2002/0156772 A1

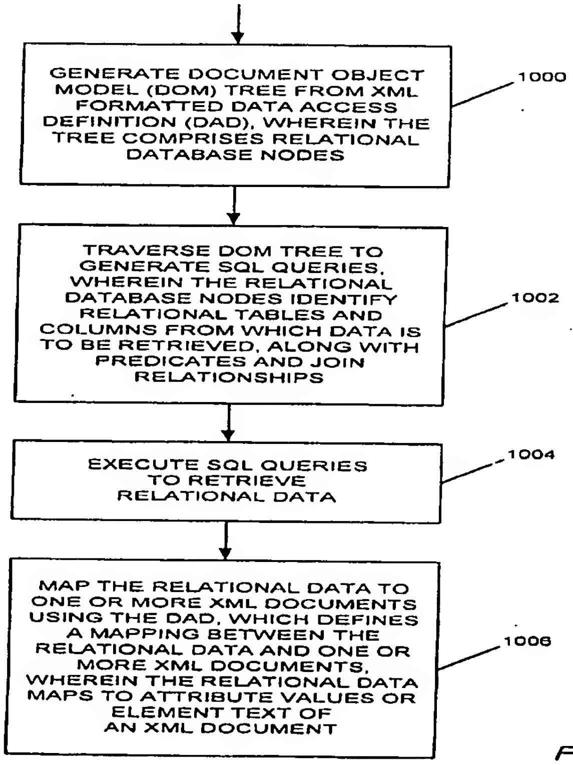
GENERATE DOCUMENT OBJECT MODEL (DOM) TREE FROM XML FORMATTED DATA ACCESS DEFINITION (DAD), WHEREIN THE TREE COMPRISES RELATIONAL DATABASE NODES
1000
TRAVERSE DOM TREE TO GENERATE SQL QUERIES, WHEREIN THE RELATIONAL DATABASE NODES IDENTIFY RELATIONAL TABLES AND COLUMNS FROM WHICH DATA IS TO BE RETRIEVED, ALONG WITH PREDICATES AND JOIN RELATIONSHIPS
1002
EXECUTE SQL QUERIES TO RETRIEVE RELATIONAL DATA
1004
MAP THE RELATIONAL DATA TO ONE OR MORE XML DOCUMENTS USING THE DAD, WHICH DEFINES A MAPPING BETWEEN THE RELATIONAL DATA AND ONE OR MORE XML DOCUMENTS, WHEREIN THE RELATIONAL DATA MAPS TO ATTRIBUTE VALUES OR ELEMENT TEXT OF AN XML DOCUMENT
1006

FIG. 10

Regarding claim 2, applicant argues that **Chau** does not teach **"the method of claim 1, including the further step of generating a collection of parent nodes that potentially have child nodes satisfying the Xpath expression from a table relating a class of parent nodes/parent nodes and a class of child nodes/child nodes, and wherein the table is used in the final searching step."**

In response to the preceding argument, Examiner respectfully submits that **Chau** teaches **"the method of claim 1, including the further step of generating a collection of parent nodes that potentially have child nodes satisfying the Xpath expression from a table relating a class of parent nodes/parent nodes and a class of child nodes/child nodes, and wherein the table is used in the final searching step"** as for the root element_node, all tables storing its attribute or all child element data should be specified (**Chau** Paragraph 0164). Representing an XML element. It must be defined in the specified DTD. For the RDB_node mapping, the root element_node must have a RDB_node to specify all tables containing XML data for itself and all its children nodes. It can have zero or more attribute_nodes and child element_nodes, as well as zero or one text_node (**Chau** Paragraph 0176).

Further, **Chau** teaches Initially, an XML document containing XML data is received. A document access definition that identifies one or more relational tables and columns is received. The XML data is mapped from the application DTD to the relational tables and columns using the document access definition based on the XPath data model (**Chau** Paragraph 0019).

Examiner interprets that root node and further child nodes could also be parent node if they also have children nodes except the last node in the hierarchy. The RDB_node defines the mapping between an XML element or attributes and relational data tables.

Regarding claims 4 and 5, applicant argues that **Chau** does not teach **"the method of claim 2, wherein the table comprises entries containing hash representations of a class of parent nodes/parent nodes and a class of child nodes/child nodes."**

In response to the preceding argument, Examiner respectfully submits that **Chau** teaches **"the method of claim 2, wherein the table comprises entries containing hash representations of a class of parent nodes/parent nodes and a class of child nodes/child nodes"** as levelmap is an associative array that maps column names to their equivalence class numbers or "relational level." The equivalence classes in ascending order of relational levels should have one-to-many relationship between each adjacent classes with the "many" side at the upper level. In the example, "order_key" maps to 0; "part_key" maps to 1; and "date" maps to 2. The associative array can be implemented in memory, for example, as a hash table, a sorted array, or a binary search tree (**Chau** Paragraph 0699). Create a hash table or stored array of all top_elements rel->top_elements for fast search (**Chau** Paragraph 0941).

Further, **Chau** teaches if these relationships are visualized in the form of a tree, order could be regarded as the root, which has parts as its children, and each part has shipments as its children. For this discussion, the different levels of this tree are called "relational levels" (Chau Paragraph 0690).

Therefore these levelmaps are relational levels in the relational tables of claim 2.

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Examiner's Note: Examiner has cited particular figures, and paragraphs in the references as applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings in the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant, in preparing the responses, to fully consider the references in entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the examiner.

Contact Information

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Usmaan Saeed whose telephone number is (571)272-4046. The examiner can normally be reached on M-F 8-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hosain Alam can be reached on (571)272-3978. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Usmaan Saeed
Patent Examiner
Art Unit: 2166

Leslie Wong LW
Primary Examiner

US
February 28, 2007s

HOSAIN ALAM
SUPERVISORY PATENT EXAMINER